IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant:	David E. LOWELL	§	Confirmation No.:	8180
Serial No.:	10/677,159	§	Group Art Unit:	2195
Filed:	October 1, 2003	§	Examiner:	Meng Yao Zhe
For:	Runtime Virtualization And Devirtualization Of Memory By A Virtual Machine Monitor	§	Docket No.:	200300561-1

APPEAL BRIEF

Mail Stop Appeal Brief – Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Date: May 17, 2010

Sir:

Appellant hereby submits this Appeal Brief in connection with the above-identified application. A Notice of Appeal was electronically filed on March 15, 2010.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, L.P. (HPDC), a Texas Limited Partnership, having its principal place of business in Houston, Texas. HPDC is a wholly owned affiliate of Hewlett-Packard Company (HPC). The Assignment from the inventor to HPDC was recorded on December 2, 2003, at Reel/Frame 014168/0718.

II. RELATED APPEALS AND INTERFERENCES

Appellant notes that patent application Serial No. 10/676,922 is on appeal and may contain subject matter that is related to the present case.

III. STATUS OF THE CLAIMS

Originally filed claims:	1-67.
Claim cancellations:	None.
Added claims:	None.
Presently pending claims:	1-67.
Presently appealed claims:	1-67.

IV. STATUS OF THE AMENDMENTS

No claims were amended after the Final Office Action dated December 22, 2010.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

This section provides a concise explanation of the subject matter defined in each of the independent claims, referring to the specification by page and line number or to the drawings by reference characters as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified with a corresponding reference to the specification or drawings where applicable. The specification references are made to the application as filed by Appellant. Note that the citation to passages in the specification or drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element. Also note that these specific references are not exclusive; there may be additional support for the subject matter elsewhere in the specification and drawings.

In accordance with the invention of claim 1, a method[1] of running a virtual machine monitor[2] on computer hardware,[3] the hardware including memory,[4] the method comprises commencing virtualization[5] of the memory at multiple times during runtime of the operating system.[6]

In accordance with the invention of claim 13, a method[7] of running a virtual machine monitor[8] on computer hardware[9] and an operating system[10] on the virtual machine monitor, the hardware including memory,[11] the memory

---

[1] Fig. 2. Disclosure p. 6 line 1 of para. [0026].

[2] Fig. 1, VMM 114. Disclosure p. 5 line 2 of para. [0024].

[3] Fig. 1, hardware 110. Disclosure p. 5 line 2 of para. [0023].

[4] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

[5] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030].

[6] Disclosure p. 5 lines 1-4 of para. [0025].

[7] Fig. 2. Disclosure p. 6 line 1 of para. [0026].

[8] Fig. 1, VMM 114. Disclosure p. 5 line 2 of para. [0024].

[9] Fig. 1, hardware 110. Disclosure p. 5 line 2 of para. [0023].

[10] Fig. 1, OS 112. Disclosure p. 5 line 1 of para. [0024].

[11] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

virtualized by the virtual machine monitor,[12] the method comprises devirtualizing the memory at runtime of an operating system.[13]

In accordance with the invention of claim 28, a computer[14] comprises memory[15] including first and second portions. The first portion is encoded with a virtual machine monitor[16] that commences virtualization of the second portion during runtime of an operating system.[17]

In accordance with the invention of claim 36, an article for a computer comprises computer memory[18] including a first portion encoded with a virtual machine monitor[19] that commences virtualization of a second portion of the memory multiple times during runtime of an operating system.[20]

In accordance with the invention of claim 44, a computer[21] comprises hardware[22] including memory[23] and a virtual machine monitor[24] for virtualizing the memory and devirtualizing the memory at runtime.[25] The virtual machine monitor

---

[12] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030].

[13] Fig. 2, 220. Disclosure p. 7 line 6 of para. [0030].

[14] Fig. 1, computer 100. Disclosure p. 5 line 1 of para. [0023].

[15] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

[16] Fig. 1, VMM 114. Disclosure p. 5 line 2 of para. [0024].

[17] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030]; p. 8 line 1 of para. [0032] through p. 12 line 4 of para. [0046].

[18] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

[19] Fig. 1, VMM 114. Disclosure p. 5 line 2 of para. [0024].

[20] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030]; p. 8 line 1 of para. [0032] through p. 12 line 4 of para. [0046].

[21] Fig. 1, computer 100. Disclosure p. 5 line 1 of para. [0023].

[22] Fig. 1, hardware 110. Disclosure p. 5 line 2 of para. [0023].

[23] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

[24] Fig. 1, VMM 114. Disclosure p. 5 line 2 of para. [0024].

[25] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030]; p. 8 line 1 of para. [0032] through p. 12 line 4 of para. [0046].

virtualizes the memory when multiple operating system instances are running and devirtualizes the memory when a single operating system instance is running.[26]

In accordance with the invention of claim 56, an article for a computer[27] including hardware,[28] the hardware including computer memory,[29] the article comprises memory encoded with software for devirtualizing the computer memory at runtime of an operating system.[30]

---

[26] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030]; p. 8 line 1 of para. [0032] through p. 12 line 4 of para. [0046].

[27] Fig. 1, computer 100. Disclosure p. 5 line 1 of para. [0023].

[28] Fig. 1, hardware 110. Disclosure p. 5 line 2 of para. [0023].

[29] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

[30] Fig. 2, 220. Disclosure p. 7 line 6 of para. [0030].

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 13-27 and 56-67 comply with the enablement requirement of 35 U.S.C. § 112, first paragraph.

Whether claims 1-12 and 28-55 comply with the written description requirement of 35 U.S.C. § 112, first paragraph.

Whether claims 1-67 are indefinite under 35 U.S.C. § 112, second paragraph.

Whether claims 1, 28, and 36 are obvious on the ground of non-statutory obviousness-type double patenting over claim 1 of co-pending U.S. patent application Serial No. 10/676,922 in view of Bean (U.S. Pat. No. 4,843,541).

Whether claims 13 and 56 are obvious on the ground of non-statutory obviousness-type double patenting over claim 17 of co-pending U.S. patent application Serial No. 10/676,922 in view of Bean.

Whether claims 1, 28, and 36 are obvious on the ground of non-statutory obviousness-type double patenting over claim 1 of co-pending U.S. patent application Serial No. 10/676,577 in view of Bean.

Whether claims 13 and 56 are obvious on the ground of non-statutory obviousness-type double patenting over claim 62 of co-pending U.S. patent application Serial No. 10/676,577 in view of Bean.

Whether claims 1, 2, 4, 5, 10, 12-19, 25, 27-30, 35-38, 42, 43, 55-62, and 67 are anticipated (35 U.S.C. § 102(b)) by Bean.

Whether claims 6-9, 11, 26, 31-34, 39-41, 44-50, 54, and 66 are obvious (35 U.S.C. § 103) over Bean.

Whether claims 20-22, 51-53, and 63-65 are obvious (35 U.S.C. § 103) over Bean in view of Chu (U.S. Pat. No. 6,256,657).

Whether claims 23-24 are obvious (35 U.S.C. § 103) over Bean in view of Waldspurger (U.S. Pat. No. 6,789,156)

Whether claim 3 is obvious over (35 U.S.C. § 103) Bean in view of Bugnion (U.S. Pat. No. 6,296,847).

VII. ARGUMENT

A. Enablement rejection of claims 13-27 and 56-67

The Examiner alleges that "virtualization is critical or essential to the practice of the invention" and failure to include the step of virtualization in claims 13-27 and 56-67 renders the claims unsupported by an enabling disclosure. 35 U.S.C. § 112, first paragraph. The Examiner notes that "memory must be virtualized before it can be devirtualized." See Final Office Action page 2.

Appellant's disclosure is directed to reciprocal processes of virtualization and devirtualization of memory. Some claims are directed to the process of virtualization and other claims are directed to the reciprocal process of devirtualization. Just because memory must be virtualized before it can be devirtualized does not mean that all claims must require virtualization. The system must also be powered on as well, but the step of powering on the system need not be included in the claim as being "essential."

B. Written description rejection of claims 1-12 and 28-55

The Examiner acknowledges that the disclosure explains what happens when virtualization is performed, but does not seem to understand what is included in the act of "commencing virtualizaton." Final Office Action page 2. The Examiner's stated confusion would seem to be more appropriate for a 35 U.S.C. § 112, second paragraph rejection, not a written description rejection. Even if the Examiner's observation is true (that commencement of virtualization is not well-explained), the Examiner has established a prima facie case for a written description rejection. At any rate, if the act of virtualization is well-explained in the disclosure, which the Examiner agrees is the case, then commencing the well-explained act would be sufficiently clear to one of ordinary skill in the art.

C. Indefiniteness rejection of claims 1-67

With regard to claim 1, the Examiner is unclear what happens when virtualization is commenced. Final Office Action page 3. This point is addressed above.

With regard to claim 13, the Examiner is uncertain how devirtualization can happen without a virtualization step. Final Office Action page 3. Just because

the memory is virtualized before it is devirtualized does not mean that the act of virtualizing memory must be included in the claims in order to claim devirtualizing memory.

D. Double patenting rejection of claims 1, 28, and 36

The Board is requested to hold this provisional double patenting rejection in abeyance pending the outcome of this present appeal or the outcome of co-pending case Serial No. 10/676,922.

E. Double patenting rejection of claims 13 and 56

The Board is requested to hold this provisional double patenting rejection in abeyance pending the outcome of this present appeal or the outcome of co-pending case Serial No. 10/676,922.

F. Double patenting rejection of claims 1, 28, and 36

Appellant notes that allegedly conflicting claim 1 of co-pending case Serial No. 10/676,557 was amended on December 22, 2009, the same day that the Final Office Action in this present case was mailed by the Office. Appellant suspects that claim 1 in co-pending case 10/676,557 was not considered in its newly amended form when issuing the present Final Office Action. The amendment further distances claim 1 in the co-pending case from the claims of this present case.

At any rate, the Board is requested to hold this provisional double patenting rejection in abeyance pending the outcome of this present appeal or the outcome of co-pending case Serial No. 10/676,557.

G. Double patenting rejection of claims 13 and 56

Applicant notes that allegedly conflicting claim 62 of co-pending case Serial No. 10/676,557 was amended on December 22, 2009, the same day that the Final Office Action in this present case was mailed by the Office. Appellant suspects that claim 62 in co-pending case Serial No. 10/676,557 was not considered by the Examiner in its newly amended form when issuing the present Final Office Action. The amendment further distances claim 62 in the co-pending case from the claims of this present case.

At any rate, the Board is requested to hold this provisional double patenting rejection in abeyance pending the outcome of this present appeal or the outcome of co-pending case Serial No. 10/676,557.

H. Anticipation rejection of claims 1, 2, 4, 5, 10, 12-19, 25, 27-30, 35-38, 42, 43, 55-62, and 67 over Bean

Claim 1 requires "commencing virtualization of the memory **at multiple times during runtime of an operating system**." The Examiner seems to believe that Bean teaches commencing virtualization during run-time of the virtual machine monitor (referred to as "VM/SP" in col. 1 lines 25-28) of Bean and that the VMM runs on top of the operating system (OS). Thus, per the Examiner's reasoning, Bean teaches commencing virtualization during run-time of the OS. Appellant disagrees. In Bean the VM/SP is a hypervisor that runs directly on the hardware and has ultimate control over the entire system. The VM/SP launches the guests and each guest runs an OS. Thus, each OS runs on top of the VM/SP. As such, it is not necessarily true that the virtualization is commenced during run-time of the OS—in fact, the OS may not even be running at that time.

Bean teaches V=V and V=F guests. A V=V guest is one that is fully virtualized. See e.g., col. 1 line 60. A V=F guest is not virtualized. See col. 8 line 24. To commence virtualization would require Bean to convert a V=F guest to a V=V guest. As noted above, claim 1 requires commencing virtualization multiple times during run-time of the operating system. Thus, Bean would have to convert a V=F guest to a V=V guest multiple times during run-time of the operating system. Bean has no such teaching. The Examiner allegedly that such a teaching is present in col. 2, lines 25-40, col. 22, lines 47-55, and col. 29, lines 35-50. Final Office Action page 10. The passages cited by the Examiner teach dynamically changing resource assignment for Bean's V=F guest. Such changing resource assignment is not at all the same as converting a V=F guest (not virtualized) to a V=V guest (fully virtualized). In fact, Bean does not teach that the preferred guest OS is even running when the alleged conversion occurs from a V=F guest to a V=V guest.

Further, Bean certainly has no teaching of commencing virtualization multiple times during run-time of the OS.

For at least these reasons, the Examiner erred in rejecting claim 1 and all claims dependent thereon over Bean. All other claims in this grouping are allowable for the same or similar reasoning. Some such claims refer to devirtualizing memory during run-time of the operating but the argument is largely the same.

I. Obviousness rejection of claims 6-9, 11, 26, 31-34, 39-41, 44-50, 54, and 66 over Bean

Claims 6-9, 11, 26, 31-34, 39-41, and 66 depend from independent claims that are allowable over Bean as explained above.

Independent claim 44 requires the VMM to commence virtualization of memory when multiple operating system instances are running and to devirtualize memory when a single operating system is running. As explained above, Bean has no teaching of converting a V=F guest to a V=V guest, and vice versa when an operating system is actually running. The OS is not described as actually being running when virtualization or devirtualization is commenced by the VM/SP. For at least this reason, the Examiner erred in rejecting claim 44 and dependent claims 45-50 and 54.

J. Obviousness rejection of claims 20-22, 51-53, and 63-65 are obvious over Bean in view of Chu

The claims in this group depend from base claims that are allowable over Bean as explained above. Chu does not satisfy the deficiencies of Bean. Thus, the claims in this group are allowable over the combination of Bean and Chu.

K. Obviousness rejection of claims 23-24 are obvious over Bean in view of Waldspurger

The claims in this group depend from base claims that are allowable over Bean as explained above. Waldspurger does not satisfy the deficiencies of Bean. Thus, the claims in this group are allowable over the combination of Bean and Waldspurger.

L. Obviousness rejection of claim 3 is obvious over Bean in view of Bugnion

Claim 3 depends from claim 1 which is allowable over Bean as explained above. Bugnion does not satisfy the deficiencies of Bean. Thus, the claims in this group are allowable over the combination of Bean and Bugnion.

M. Conclusion

For the reasons stated above, Appellant respectfully submits that the Examiner erred in rejecting all pending claims. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's Deposit Account No. 08-2025.

Respectfully submitted,

/Jonathan M. Harris/

Jonathan M. Harris
PTO Reg. No. 44,144
CONLEY ROSE, P.C.
(713) 238-8000 (Phone)
(713) 238-8008 (Fax)
ATTORNEY FOR APPELLANT

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Dept., M/S 35
3404 E. Harmony Road
Fort Collins, CO 80528-9599

VIII. CLAIMS APPENDIX

1. A method of running a virtual machine monitor on computer hardware, the hardware including memory, the method comprising:

commencing virtualization of the memory at multiple times during runtime of an operating system.

2. The method of claim 1, wherein the virtualization includes constructing an Identity mapping of physical to machine memory; and commencing to use the virtual machine monitor at runtime to manage memory translation.

3. The method of claim 2, wherein the Identity mapping is constructed prior to runtime.

4. The method of claim 2, wherein the memory translation is initially performed according to the Identity mapping.

5. The method of claim 4, wherein the virtual machine monitor modifies the mapping after the physical memory has been virtualized.

6. The method of claim 2, wherein the memory translation is managed by allowing the operating system to define virtual-to-physical mapping, and the virtual machine monitor to define physical-to machine mapping.

7. The method of claim 6, wherein the virtual machine monitor dynamically composes virtual-to-physical translations with physical-to-machine translations.

8. The method of claim 6, wherein the virtual machine monitor inspects the virtual-to-physical mappings by the operating system and maintains page tables of virtual-to-machine mappings.

9. The method of claim 6, further comprising loading a translation lookaside buffer with virtual-to-machine translations.

10. The method of claim 1, wherein only a portion of the memory is virtualized at runtime.

11. The method of claim 1, wherein the hardware includes a CPU that was virtualized prior to the virtualization of the memory.

12. The method of claim 1, further comprising performing runtime devirtualization of the virtualized memory.

13. A method of running a virtual machine monitor on computer hardware and an operating system on the virtual machine monitor, the hardware including memory, the memory virtualized by the virtual machine monitor, the method comprising:

devirtualizing the memory at runtime of an operating system.

14. The method of claim 13, wherein a portion of the memory is devirtualized.

15. The method of claim 13, wherein when the operating system is booted, the virtual machine monitor exposes the booting operating system to physical memory no larger than machine memory, where the physical memory does not span any memory holes.

16. The method of claim 13, wherein the operating system defines virtual-to-physical translations prior to the runtime devirtualization; wherein the virtual machine monitor defines physical-to-machine translations prior to the runtime devirtualization; wherein the virtual machine monitor composes dynamically the virtual-to-physical translations with the physical-to-machine translations prior to the runtime devirtualization, wherein the runtime devirtualization includes having the virtual machine monitor cease to perform the dynamic composition of translations.

17. The method of claim 13, wherein the devirtualization includes remapping physical memory so a physical-to-machine mapping becomes an Identity mapping; and using the operating system to manage address translation with respect to the devirtualized memory.

18. The method of claim 17, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped directly.

19. The method of claim 17, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped indirectly.

20. The method of claim 17, wherein the remapping of the physical memory is performed concurrently with operating system and application activity.

21. The method of claim 20, further comprising preventing the physical-to-machine mapping from being modified during the remapping, and temporarily preventing some or all write accesses to memory.

22. The method of claim 17, wherein the operating system and any application activity is paused while the remapping is performed.

23. The method of claim 17, further comprising maintaining a back map that contains for each page of machine memory a list of the pages of physical memory that map to it, and a list of free machine pages.

24. The method of claim 17, wherein the remapping is performed without a back map by maintaining a reference count for each machine page is kept, and freeing machine pages when their reference counts are zero.

25. The method of claim 17, wherein the remapping is performed without a back map by constructing a list of the physical pages mapping to a page of machine memory by searching the physical-to-machine mapping.

26. The method of claim 17, wherein managing the address translation includes having the virtual machine monitor cease to inspect the operating system's virtual-to-physical translations; and ceasing to maintain a page table of direct virtual-to-machine mappings.

27. The method of claim 17, wherein managing the address translation includes having the virtual machine monitor cease to compose dynamically the operating system's virtual-to-physical translations with the virtual machine monitor's physical-to-machine translations for a portion of physical memory that is devirtualized.

28. A computer comprising memory including first and second portions, the first portion encoded with a virtual machine monitor that commences virtualization of the second portion multiple times during runtime of an operating system.

29. The computer of claim 28, wherein the virtualization includes constructing an Identity mapping of physical to machine memory; and commencing to use the virtual machine monitor at runtime to manage memory translation.

30. The computer of claim 29, wherein the virtual machine monitor modifies the mapping after the physical memory has been virtualized.

31. The computer of claim 29, wherein an operating system is running on the virtual machine monitor prior to virtualizing the memory; and wherein the memory translation is managed by allowing the operating system to manage virtual-to-physical mapping, and allowing the virtual machine monitor to manage physical-to machine mapping.

32. The computer of claim 31, wherein the virtual machine monitor dynamically composes virtual-to-physical translations with the physical-to-machine translations.

33. The computer of claim 31, wherein the virtual machine monitor inspects the virtual-to-physical mappings by the operating system and maintains page tables of virtual-to-machine mappings.

34. The computer of claim 31, wherein a translation lookaside buffer is loaded with the virtual-to-machine translations.

35. The computer of claim 28, wherein only a portion of physical memory is virtualized at runtime.

36. An article for a computer, the article comprising computer memory including a first portion encoded with a virtual machine monitor that commences virtualization of a second portion of the memory multiple times during runtime of an operating system.

37. The article of claim 36, wherein the virtualization includes constructing an Identity mapping of physical to machine memory; and commencing to use the virtual machine monitor at runtime to manage memory translation.

38. The article of claim 37, wherein the virtual machine monitor can modify the mapping after the physical memory has been virtualized.

39. The article of claim 37, wherein the memory translation is managed by allowing an operating system to manage virtual-to-physical mapping, and allowing the virtual machine monitor to manage physical-to machine mapping.

40. The article of claim 39, wherein the virtual machine monitor can dynamically compose virtual-to-physical translations with the physical-to-machine translations.

41. The article of claim 39, wherein the virtual machine monitor can inspect the virtual-to-physical mappings by the operating system and maintains page tables of virtual-to-machine mappings.

42. The article of claim 37, wherein the virtual machine monitor can load a translation lookaside buffer with virtual-to-machine translations.

43. The article of claim 36, wherein the virtual machine monitor can virtualize only a portion of physical memory at runtime.

44. A computer comprising hardware including memory; and a virtual machine monitor for virtualizing the memory and devirtualizing the memory at runtime, wherein the virtual machine monitor commences virtualization of the memory when multiple operating system instances are running and devirtualizes the memory when a single operating system instance is running.

45. The computer of claim 44, wherein a portion of the memory is devirtualized.

46. The computer of claim 44, wherein when an operating system is booted, the virtual machine monitor exposes the booting operating system to physical memory no larger than machine memory, where the physical memory does not span any memory holes.

47. The computer of claim 44, wherein an operating system defines virtual-to-physical translations prior to the runtime devirtualization; wherein the virtual machine monitor defines physical-to-machine translations prior to the runtime devirtualization; wherein the virtual machine monitor composes dynamically the virtual-to-physical translations with the physical-to-machine translations prior to the runtime devirtualization; wherein the runtime devirtualization includes having the virtual machine monitor cease to perform the dynamic composition of translations.

48. The computer of claim 44, wherein the devirtualization includes remapping physical memory so a physical-to-machine mapping becomes an Identity mapping; and using an operating system to manage address translation with respect to the devirtualized memory.

49. The computer of claim 48, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped directly.

50. The computer of claim 48, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped indirectly.

51. The computer of claim 48, wherein the remapping of the physical memory is performed concurrently with operating system and application activity.

52. The computer of claim 51, wherein the physical-to-machine mapping is prevented from being modified during the remapping, and some or all write accesses to memory are temporarily prevented.

53. The computer of claim 48, wherein the operating system and any application activity is paused while the remapping is performed.

54. The computer of claim 48, wherein managing the address translation includes having the virtual machine monitor cease to inspect the operating system's virtual-to-physical translations; and wherein maintenance of a page table of direct virtual-to-machine mappings is ceased.

55. The computer of claim 48, wherein managing the address translation includes having the virtual machine monitor cease to compose dynamically the operating system's virtual-to-physical translations with the virtual machine monitor's physical-to-machine translations for a portion of physical memory that is devirtualized.

56. An article for a computer including hardware, the hardware including computer memory, the article comprising memory encoded with software for devirtualizing the computer memory at runtime of an operating system.

57. The article of claim 56, wherein the software causes a portion of the memory to be devirtualized.

58. The article of claim 56, wherein the software includes a virtual machine monitor; and wherein when an operating system is booted on the virtual machine monitor, the virtual machine monitor exposes the booting operating system to physical memory no larger than machine memory, where the physical memory does not span any memory holes.

59. The article of claim 56, wherein an operating system defines virtual-to-physical translations prior to the runtime devirtualization; wherein the software includes a virtual machine monitor for defining physical-to-machine translations prior to the runtime devirtualization, composing dynamically the virtual-to-physical

translations with the physical-to-machine translations prior to the runtime devirtualization, and ceasing to perform the dynamic composition of translations during the runtime virtualization; and wherein after the runtime devirtualization is performed, memory translation is performed by directly using the virtual-to-physical mapping defined by the operating system.

60. The article of claim 56, wherein the devirtualization includes remapping physical memory so a physical-to-machine mapping becomes an Identity mapping; and using an operating system to manage address translation with respect to the devirtualized memory.

61. The article of claim 60, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped directly.

62. The article of claim 60, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped indirectly.

63. The article of claim 60, wherein the remapping of the physical memory is performed concurrently with operating system and application activity.

64. The article of claim 63, wherein the physical-to-machine mapping is prevented from being modified during the remapping, and some or all write accesses to memory are temporarily prevented.

65. The article of claim 60, wherein the operating system and any application activity is paused while the remapping is performed.

66. The article of claim 60, wherein the software includes a virtual machine monitor that manages the address translation by ceasing to inspect the operating system's virtual-to-physical translations; and wherein maintenance of a page table of direct virtual-to-machine mappings is ceased.

67. The article of claim 60, wherein the software includes a virtual machine monitor for managing the address translation by ceasing to compose dynamically the operating system's virtual-to-physical translations with the virtual machine monitor's physical-to-machine translations for a portion of physical memory that is devirtualized.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.